

一种改进的基于过度分割的图像彩色化方法

滕升华^{①②} 湛安军^{①②} 袁建华^{①②} 殷学民^{①②} 王东峰^① 邹谋炎^①

^①(中国科学院电子学研究所 北京 100080)

^②(中国科学院研究生院 北京 100039)

摘要 黑白图像的彩色化是图像处理领域一个活跃的、有挑战性的研究课题。该文在总结图像彩色化方法的基础上,重点分析了Levin方法;针对这种方法处理大尺寸图像速度较慢的问题,提出先对图像进行过度分割,再以区域为基本单元进行彩色化的方案。实验表明,该方法能够在几乎不影响彩色化效果的前提下,显著提高处理大尺寸图像的速度。

关键词 黑白图像,彩色化,过度分割

中图分类号:TN911.73

文献标识码:A

文章编号:1009-5896(2006)07-1160-05

An Improved Colorization Algorithm for Gray-Scale Image Based on Over-Segmentation

Teng Sheng-hua^{①②} Chen An-jun^{①②} Yuan Jian-hua^{①②}

Yin Xue-min^{①②} Wang Dong-feng^① Zou Mou-yan^①

^①(Institute of Electronics, Chinese Academy of Sciences, Beijing 100080, China)

^②(The Graduate School of Chinese Academy of Sciences, Beijing 100039, China)

Abstract Colorization of gray-scale image is in general an active and challenging area of research in image processing. In this paper current colorizing algorithms are summarized, of which Levin's method is analyzed in particular. According to the speed dropping of Levin's method in processing large scale images, a new scheme based on over-segmentation is proposed. Experiments show that the new scheme can remarkably quicken the processing of large scale images, nearly without degrading the colorizing effect.

Key words Gray-scale image, Colorization, Over-segmentation

1 引言

彩色化是一个给黑白图像、电影或电视节目加上颜色的处理过程,一般认为是由Markle在1970年发明的,最初用于处理阿波罗登月计划获取的月球影像^[1]。

黑白电影的彩色化在上世纪80年代曾经是一个热门话题,一些早期电影通过彩色化处理焕发了新的光彩。尽管人们对此一直颇有争议,但时至今日,彩色化研究仍然是图像处理学界一个活跃的、有挑战性的课题,更多地应用于图像、视频编辑和图像通信等领域^[2]。

2 图像彩色化的研究进展

这里所说的彩色化处理要求给黑白图像赋予自然的、接近于真实的色彩,严格讲是一种假彩色处理^[3]。与医学、遥感等领域广泛应用的伪彩色处理的不同之处在于,假彩色化力求再现影像场景的本来面目,而不仅仅是为了使图像内容更加醒目,后者可以完全不顾忌图像中真实的颜色。

图像彩色化从根本上来讲就是要对图像中的每个像素,用多维颜色空间的一个矢量(如表示亮度、色度、饱和度的三维颜色空间)取代表示亮度的一维标量;这显然是一个

“欠约束”问题,或者称“病态”问题。因此,图像彩色化没有所谓唯一正确的结果,而且目前缺乏评价彩色化结果的客观标准。要解决这个病态问题,需要利用某些先验知识、设定一些合理的约束。

电影的彩色化处理,一般先由专门的画师对某些特定的帧图像进行手工着色;这种手工制作方式不仅对操作者自身水平要求较高,而且效率低下。近年来图像彩色化技术有了新的发展,出现了一些半自动的处理方法,大致可归结为两类,即基于颜色转移的彩色化和局部彩色化向全局扩展的方法。

2.1 基于颜色转移的彩色化

图像彩色化的一大进步首先来源于Welsh等人的工作,通过将彩色参考图像上的颜色转移到黑白图像上,实现了一种半自动化的图像彩色化技术^[4]。具体做法是考查黑白图像上每个像素的一个邻域,在彩色参考图像上寻找一个与之在亮度上相匹配的区域,然后将找到的匹配区域中心的颜色值作为当前像素的彩色化结果。这种彩色化方法尤其适用于这样的图像,即图像上颜色不同的区域对应不同的亮度或者表现出不同的纹理特征;除此之外,必须手工指定黑白图像与彩色参考图像之间的匹配区域。

尽管文献[4]给出了一些效果不错的彩色化实例,但是这类方法不能保证处理后的颜色在空间上的连续性,即本来相邻的、颜色相近的区域可能由于亮度差异而被分配了截然不同的颜色。同时,这种方法的彩色化结果强烈依赖于所采用的参考图像,这使得参考图像的选择成了一个新的难题;由此带来的另一个问题是对彩色化结果的事先控制和局部调整都非常不便,而这一点往往又是很重要的。

2.2 局部彩色化的全局扩展

前已提及,图像彩色化是一个病态问题。由于这种病态性,基于颜色转移的彩色化方法要借助于额外的彩色图像作为选色参考;与之对应的另外一种考虑是,先对图像的部分像素利用某种方法进行粗略的彩色化,再结合图像的自身特征设法将局部的彩色化结果扩展到整幅图像。

基于这种考虑, Horiuchi 等人在图像彩色化方面做了一些工作^[5-8]。文献[5,6]假定,图像的颜色值服从局部的 Markov 性,即图像中一点的颜色仅与其邻近像素有关。基于此,设定一些事先彩色化的“种子像素”,把图像中所有像素与其邻近像素颜色差异的总和作为最小化目标,将图像彩色化转换为一个极值问题,计算过程采用随机松弛优化算法。

诚然,也正如文献[5,6]中所见,这种总体优化的方法可以在某种程度上解决图像彩色化问题,但其缺点也是相当明显的,除了处理效果差,仅仅是巨大的计算量就妨碍了该方法的实际应用。为了提高处理速度,文献[7,8]采用了由种子像素开始、在保证邻域像素颜色差异最小的前提下向外逐点扩张的彩色化方法;为改善边界的彩色化效果引进了边界划分算法,并且分别在 RGB 和 CIELAB 颜色空间实施了该算法。与利用随机松弛法进行总体优化相比,这种逐点扩张的彩色化方法能够显著提高处理速度,但文中采用的种子像素是从初始彩色图像中随机抽取的,这在实际的图像彩色化中很难实施。

从类似的前提出发, Levin 等人的做法是先由人工在图像中各个区域涂上适当的彩色线条,颜色扩展过程要使所有像素与其邻域像素色度加权差的平方和最小。由于颜色扩展是在色差空间(YUV 颜色空间)进行的,不仅能保证图像在处理前后亮度不变,而且消除了直接在 RGB 空间进行处理可能产生的颜色混扰。辅之以适当的人工着色点(彩色线条), Levin 颜色扩展方法可以取得相当不错的彩色化效果^[9]。

综合对比来看, Levin 的颜色扩展方法简易可行且效果颇佳,对于尺寸较小的图像其处理速度也是不错的,但是处理速度随着图像尺寸增大显著下降:采用同样的程序,在同一台 PC 机上处理 128×128 的图像耗时 4.3s,处理 256×256 的图像耗时 21.5s,对于 512×512 的图像其处理时间竟长达 150s 以上,而且对于稍大的图像在处理中很容易造成计算机

内存溢出。本文正是致力于提高大尺寸图像的彩色化速度,减少计算过程的内存开销,在 Levin 方法的基础上,提出一种基于多区域过度分割的彩色化方法。

3 基于过度分割的彩色化

3.1 Levin 解法的一种推导

Levin 认为,图像中相邻像素,如果它们的亮度值相近,那么它们应该具有相近的颜色值^[9]。为此,在将颜色进行 YUV 分解之后的 UV 色差空间,设定最小化目标

$$J(x) = \sum_r \left(x(r) - \sum_{s \in N(r), s \neq r} w_{rs} x(s) \right)^2 \quad (1)$$

其中 $x(r)$, $x(s)$ 表示是像素的色差, $N(r)$ 表示 r 的邻域, w_{rs} 是 r , s 间依据亮度值确定的权值,满足

$$\left. \begin{aligned} w_{rs} &\propto e^{-\frac{(Y(r)-Y(s))^2}{2\sigma_r^2}} \\ \sum_{s \in N(r), s \neq r} w_{rs} &= 1 \end{aligned} \right\} \quad (2)$$

$Y(r)$, $Y(s)$ 是像素的亮度, σ_r^2 是 r 邻域内亮度的方差;邻域外像素与邻域中心像素间的权值为 0。

式(1)难于直接进行最小化,为此将图像矩阵按列堆积表示为向量形式 $\mathbf{x}^T = [x_1, x_2, \dots, x_n]$, n 表示图像的像素总数;式(1)是向量 2-范数的平方形式,又可表示为

$$J(\mathbf{x}) = (\mathbf{Ax})^T \mathbf{Ax} = \mathbf{x}^T \mathbf{A}^T \mathbf{Ax} \quad (3)$$

其中 $\mathbf{A} = \begin{bmatrix} 1 & -w_{12} & \dots & -w_{1n} \\ -w_{21} & 1 & \dots & -w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -w_{n1} & -w_{n2} & \dots & 1 \end{bmatrix}_{n \times n}$ 。权值矩阵 \mathbf{A} 表示像素

间由亮度值确立的色差对比关系, $\mathbf{A}^T \mathbf{A}$ 与之意义一致只是取值不同,为降低运算量用 \mathbf{A} 代替式(3)中的 $\mathbf{A}^T \mathbf{A}$ 不会改变问题的本质,于是有

$$J(\mathbf{x}) = \mathbf{x}^T \mathbf{Ax} \quad (4)$$

仅从黑白图像开始进行彩色化是一个病态问题,为此在黑白图像的某些区域涂上适当的颜色作为彩色化的导引或称之为限制,彩色化的最终结果要使涂色像素的颜色保持不变。在色差空间,将涂色图像矩阵按列堆积表示为 $\mathbf{b}^T = [b_1, b_2, \dots, b_n]$, \mathbf{b} 中仅涂色位置的元素取非零值,于是有

$$\mathbf{b}^T \mathbf{x} = \mathbf{b}^T \mathbf{b} \quad (5)$$

至此,彩色化演化为在式(5)约束下对式(4)的最小化问题,可以用 Lagrange 乘子法来解决,即最小化

$$P(\mathbf{x}) = \mathbf{x}^T \mathbf{Ax} + \lambda (\mathbf{b}^T \mathbf{b} - \mathbf{b}^T \mathbf{x}) \quad (6)$$

上式对 \mathbf{x} 求导并令导数为 0,整理得

$$\mathbf{Ax} = \lambda \mathbf{b} / 2 \quad (7)$$

考虑到要使着色像素的颜色保持不变,为计算简便对

A 做一些修正: 如果图像的第 i 个像素已着色, 要使 $x_i = b_i$, 为此令 A 中 $w_{ij} = 0$ ($1 \leq j \leq n, j \neq i$), 即 A 中第 i 行元素除主对角线外都取 0, 同时只有取 $\lambda = 2$ 才能满足 $x_i = b_i$; 对 A 中其余着色像素对应行做相同处理。修正后的 A 记为 A_r , 进而有

$$A_r x = b \quad (8)$$

这样, 为图像彩色化而引出的优化问题最终归结为一个常规的矩阵方程求解问题。

3.2 改进的基于过度分割的彩色化算法

由以上分析, 为求解式(8), 需要存储一个尺寸为 $n \times n$ 的权值矩阵 A_r , 当图像尺寸较大时 A_r 将占据很大的存储空间; 同时也如前所述, 对于大尺寸图像, 应用该算法的处理时间不能令人满意。本质上讲, Levin 算法将像素作为彩色化的基本单元, 图像尺寸增大势必显著增大存储空间、延长处理时间。

实际上图像内容在一个足够小的区域上无显著变化, 相应可以假定在小的区域内图像的色差恒定, 这样就可以将“小的区域”而不是“像素”作为彩色化的基本单元。区域划分的尺寸影响彩色化的精细程度, 所以这里先对图像进行“过度分割”以得到足够小的区域, 从而保证在提高处理速度的前提下降低对彩色化结果的影响。

用区域取代像素作为彩色化的基本单元, 上面诸式(式 1-式 8)中的色差值变为区域均值, 像素间权值变更为由各区域的亮度均值决定的区域间权值, 相应的像素邻域变为区域的邻域。如此一来, 如果将图像分割成 m 个区域, 权值矩阵 A_r 的尺寸锐减为 $m \times m$ 。

4 实验结果

4.1 改进算法的一个彩色化实例

图像分割方法众多, 但至今仍然没有适用于各种场合的通用算法。实验中采用分水岭算法^[10], 该算法边界定位较好, 而且易于将图像过度分割。

图1所示为一个黑白图像彩色化的例子。图1(a)为给定的黑白图像; 图1(b)为应用分水岭算法对图1(a)进行分割的结果, 图中共划分了426个区域; 图1(c)是手工绘制的涂色图像, 用以指定各部分希望的颜色; 图1(d)是应用本算法借助图1(c)对原图进行彩色化的结果。

4.2 与原算法的比较

为了与Levin算法进行比较, 在同一台PC机、相同的编程环境下, 利用相同的涂色图像(涂色取原始彩色图像中对应位置的颜色)对同一幅黑白图像进行彩色化, 结果如图2所示。直观上看, 这两种方法的处理结果都非常接近于原始图像; 相对而言本文提出的改进算法在某些边缘附近存在一点偏差, 这是由不精确分割造成的, 研究准确快速的分割算法是完善本算法的有效途径。

为了量化彩色化图像与原始图像的近似程度, 采用信噪比峰值PSNR (Peak Signal-Noise Ratio)作为衡量指标。两幅图像间的PSNR定义^[7]为

$$PSNR = 10 \lg \frac{3n \times 255 \times 255}{\sum [(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2]} \quad (9)$$

其中 n 是图像的像素总数, r, g, b 表示像素值的RGB分量, 各分量采用8位量化。

两种算法对同一幅图像的处理时间及PSNR比较见表1。可见, 改进算法用来处理尺寸为512×512的图像能够取得比原算法快1倍的速度, 当然处理速度与图像分割的区域数目有关, 好的分割算法可以在保证分割精度的前提下减少分割区域, 自然能够进一步加快彩色化进度。两种算法的彩色化结果相对于原图的PSNR接近, 两种处理结果之间的PSNR高达32.23, 这充分反映了两种算法的处理结果及原图三者之间的高度相似性。

表 1 处理时间与 PSNR 对比
Tab.1 Comparison of processing time and PSNR

	图像尺寸	处理时间(s)	信噪比峰值(dB)
Levin 算法	512×512	150.8	27.30
改进算法	512×512	73.8	27.47

实验程序是用MALAB编写的, 如果改用C/C++, 处理速度会更快。另外, 不必对图像的每个细小区域都涂色, 算法会自动确定每个涂色区的实际作用范围, 并将颜色扩展到没有涂色的区域; 涂色要保证细小区域内颜色一致, 而实际图像的颜色也确实如此。

5 结束语

本文深入分析了 Levin 利用总体优化的图像彩色化方法, 提出以区域取代像素作为彩色化的基本单元, 改进了原算法。由于颜色扩展是在色差空间进行的, 虽然假定在一个小的区域内色差相同, 但像素间的亮度差异使得各点颜色值又不尽相同, 同时考虑到人眼对颜色的分辨能力, 可以认为本算法所作的假设是完全合理的。

图像分割结果对彩色化进程及结果影响很大: 分割精度越高、分割区域越多彩色化结果越好, 处理速度主要受分割区域数目的影响; 在保证分割精度的前提下, 适当减少分割区域几乎不影响彩色化效果。因此, 本方法的进一步完善在于寻求更加精确快速的分割算法。

处理小尺寸图像, 与原算法相比并无速度优势, 甚至还慢于原算法; 配以精确快速的图像分割, 改进算法能明显加快大尺寸图像的处理速度。因此实际处理中, 要根据图像的尺寸灵活选用基于像素的原始算法或基于多区域过度分割的改进算法。

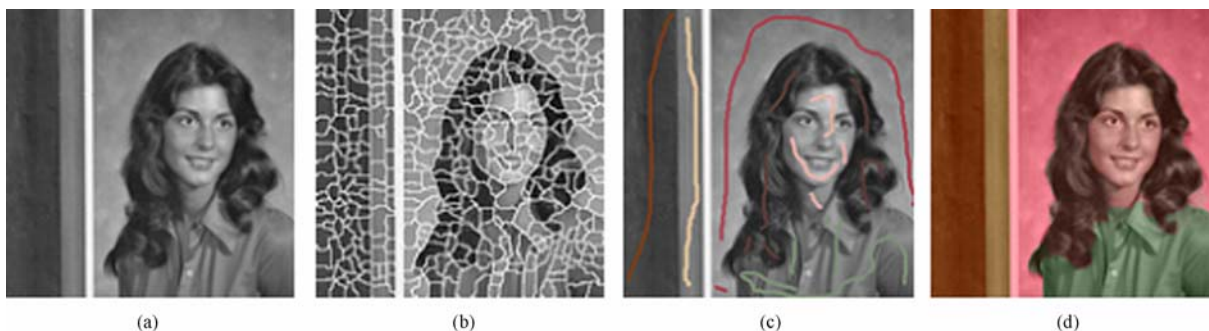


图1 黑白图像的彩色化 (a)黑白图像 (b)分割结果 (c)涂色图像 (d)彩色化效果
 Fig.1 Colorization of grayscale image (a) Grayscale image (b) Segmentation (c) Color-scribbled image (d) Colorized result

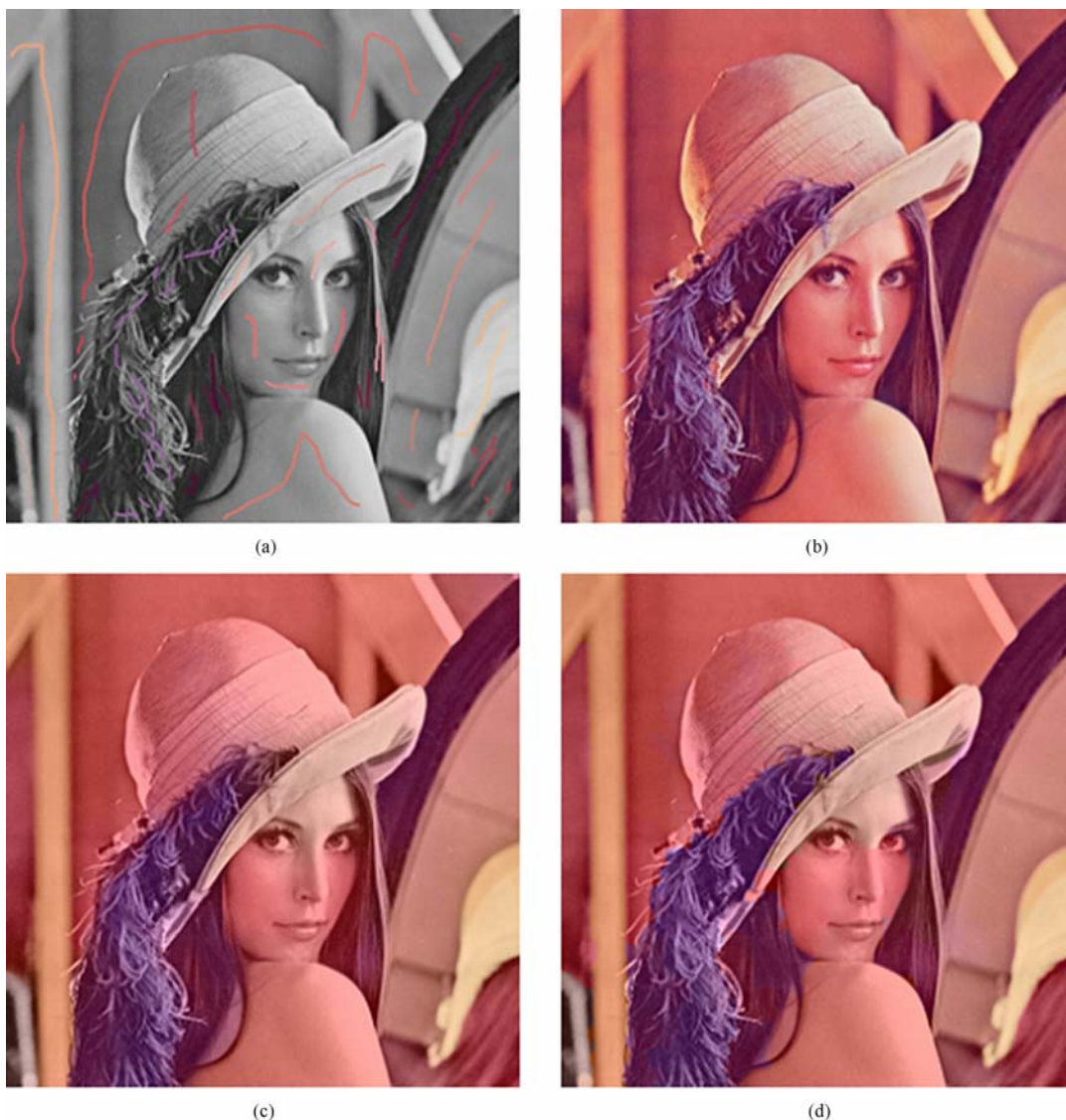


图2 改进算法与Levin算法的处理结果对比 (a) 涂色图像 (b) 原始彩色图像 (c) Levin算法的处理结果 (d) 改进算法的处理结果
 Fig.2 Comparison of visual quality with Levin's method (a) Color-scribbled image (b) Original color image
 (c) Colorized image by Levin's method (d) Colorized image by improved method

参 考 文 献

[1] Burns G. Colorization. <http://www.museum.tv/archives/etv/C/htmlC/colorization/colorization.htm>.
 [2] Sapiro G. Inpainting the colors. <http://www.ima.umn.edu/reprints/may2004/1979.pdf>.

[3] 阮秋琦. 数字图像处理学. 北京: 电子工业出版社, 2001: 212-214.
 [4] Welsh T, Ashikhmin M, Mueller K. Transferring color to grayscale images. Proc. ACM SIGGRAPH, San Antonio, USA, 2002: 277-280.

- [5] Takahiko Horiuchi. Estimation of color for gray-level image by probabilistic relaxation. Proc. IEEE 16th International Conference on Pattern Recognition, Quebec, Canada. 2002, (3): 867–870.
- [6] Takahiko Horiuchi. Colorization algorithm using probabilistic relaxation. *Image and Vision Computing*, 2004, 22(3): 197–202.
- [7] Takahiko Horiuchi, Sayaka Hirano. Colorization algorithm for grayscale image by propagating seed pixels. Proc. International Conference on Image Processing, Barcelona, Spain. 2003: 457–460.
- [8] Tomohisa Takahama, Takahiko Horiuchi, Hiroaki Kotera. Improvement on colorization accuracy by partitioning algorithm in CIELAB color space. The fifth Pacific – Rim Conference on Multimedia, Tokyo, Japan, 2004: 794–801.
- [9] Levin A, Lischinski D, Weiss Y. Colorization using optimization. Proc. ACM SIGGRAPH, Los Angeles, USA, 2004: 689–693.
- [10] Vincent L, Soille P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1991, 13(6): 583–598.
- 滕升华: 男, 1977 年生, 博士生, 研究方向为图像、视频处理技术。
- 邹谋炎: 男, 1941 年生, 研究员, 博士生导师, 研究方向为信号处理、图像处理理论和算法、无线宽带通信技术。